



Canada  
Health  
Infoway

Inforoute  
Santé  
du Canada

# **Canadian Electronic Drug (CeRx) Messaging Standard**

## **Data Types**

**October 25, 2007**

## Document Information

<b>Author:</b>	Lloyd McKenzie
<b>Creation Date:</b>	December 8, 2004
<b>Last Updated:</b>	October 25, 2007
<b>Language:</b>	English
<b>Document Number:</b>	PN502-0006-EN
<b>Document Status</b>	In Progress
<b>Infoway Project</b>	PN502 (Rx5) / CeRx
<b>Distribution</b>	Drug Standards Working Group; Other Stakeholders / Reviewers;
<b>Contact Information</b>	Toronto Office: 150 King Street West, Suite 1308 Toronto, Ontario M5H 1J9 Tel.: (416) 979-4606 Toll free: 1-888-733-6462 Fax: (416) 593-5911 <a href="http://www.Infoway-inforoute.ca">http://www.Infoway-inforoute.ca</a>

## Version Tracking

<b>Version</b>	<b>Author(s)</b>	<b>Change Description</b>	<b>Date</b>
1.00	Lloyd McKenzie	Original version and framework	2004-Dec-03
1.01	Marc Koehn	Incorporated into package publication framework.	2004-Dec-09
1.02	Lloyd McKenzie	Substantial updates (made without tracking due to the extent of the changes)	2005-Sept-11
1.03	Lloyd McKenzie	Minor update to 4.3 Coded Value	2007-July-12
1.04	Garry Cruickshank	Add QS and TRC to the NULL flavor	2007-10-25

## Acknowledgments


N/A

## Copyright Notice

This document is fully copyright protected by the owner. The owner has the exclusive right to make copies of this document. No alterations, deletions or substitutions may be made in it without the prior written consent of the owner. No part of it may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, email or any information storage and retrieval system, without the prior written consent of the owner.

## PREFACE – QUICK DOCUMENT OVERVIEW

---

<b>Purpose</b>	To summarize data types in use in the CeRx specification.
<b>Audience</b>	CeRx-pCSG (Standards Working Group) and other interested stakeholders.
<b>Instructions</b>	Instructions have been provided for each element in this document. Use the Show/Hide button in Word to show the hidden text. 

<b>Structure</b>	In addition to any “Appendix” sections, this document includes the following specific sections:	
	<b>Introduction</b>	Introduction to the Package Overview.
	<b>Data Types</b>	Describes data type constraints.

<b>Related documents</b>	<ul style="list-style-type: none"><li>• Infoway Style Guide – June 17, 2003</li></ul>
--------------------------	---

# TABLE OF CONTENTS

---

<b>I</b>	<b>Introduction</b>	<b>7</b>
1.1	Scope	7
1.2	Purpose	7
1.3	Audience	7
1.4	Assumptions	7
1.5	Related Documents	7
<b>II</b>	<b>General Notes</b>	<b>9</b>
2.1	Datatype Flavors	9
2.2	Null Flavor	9
2.3	Validation	9
<b>III</b>	<b>Collections</b>	<b>10</b>
3.1	Sets	10
3.2	Lists	10
<b>IV</b>	<b>Codes and Identifiers</b>	<b>11</b>
4.1	II - Installer Identifier	11
4.1.1	Root	11
4.1.2	Extension	11
4.2	CS - Coded Simple	11
4.2.1	Code	11
4.3	Coded Value	11
4.3.1	code	12
4.3.2	codeSystem	12
4.3.3	originalText	12
<b>V</b>	<b>Text</b>	<b>13</b>
5.1	ST - String	13
5.2	ED.DOCORREF – Encapsulated Data (Document or Reference)	13
5.2.1	compression	13
5.2.2	language	13
5.2.3	mediaType	13
5.2.4	reference	13
5.2.5	<content>	14
5.3	ED.REF – Encapsulated Data (Reference)	14
5.3.1	language	14
5.3.2	mediaType	14
5.3.3	reference	14
5.4	ED.SIGNATURE – Encapsulated Data (Signature)	14
5.4.1	mediaType	15
5.4.2	<content>	15
<b>VI</b>	<b>Timing</b>	<b>16</b>
6.1	TS.DATE – Timestamp (Date, partial allowed)	16
6.2	TS.FULLDATE – Timestamp (fully-specified date only)	16
6.3	TS.FULLDATETIME – Timestamp (fully-specified date and time only)	16
6.4	IVL.LOW<TS. DATE> – Start Date	16
6.5	IVL.LOW<TS. FULLDATE> – Start Date (complete)	16

6.6	IVL.HIGH<TS.FULLDATE> – End Date .....	17
6.7	IVL<TS. DATE> – Uncertain date range.....	17
6.7.1	low.....	17
6.7.2	high .....	17
6.7.3	center .....	17
6.7.4	width.....	17
6.8	IVL<TS. FULLDATE> – Date range.....	18
6.8.1	low.....	18
6.8.2	high .....	18
6.8.3	center .....	18
6.8.4	width.....	18
6.9	IVL.WIDTH<TS. FULLDATE> – Duration.....	18
6.9.1	width.....	19
6.10	GTS.BOUNDEDPIVL – General Timing Specification (Bounded periodic interval) .....	19
<b>VII Measurement.....</b>		<b>20</b>
7.1	INT.NONNEG – Integer (Non-negative) .....	20
7.2	INT.POS – Integer (Positive).....	20
7.3	PQ.BASIC – Physical Quantity (Basic) .....	20
7.3.1	value.....	20
7.3.2	unit .....	20
7.4	PQ.DRUG – Physical Quantity (drugs) .....	20
7.5	PQ.HEIGHTWEIGHT – Physical Quantity (height or weight) .....	20
7.6	PQ.TIME – Physical Quantity (time) .....	20
7.7	IVL<PQ.x> - Physical Quantity Range .....	21
7.8	URG<PQ.x> - Physical Quantity Uncertain Range.....	21
7.9	RTO<PQ.DRUG, PQ.TIME> Drug Quantity Per Time.....	21
<b>VIII Demographic.....</b>		<b>22</b>
8.1	PN.BASIC.....	22
8.1.1	Name use.....	22
8.1.2	Name qualifier .....	22
8.2	TN.....	22
8.3	AD.BASIC.....	22
8.4	TEL.PHONEMAIL .....	23
8.5	TEL.URI.....	23
<b>IX Other datatypes.....</b>		<b>25</b>
9.1	BL.....	25

# I INTRODUCTION

---

In HL7 version 3, the attributes of all classes are defined using a set of datatypes specific to the HL7 v3 standard. These datatypes are defined and balloted as part of the HL7 standard. The documentation on the available datatypes, their properties and their relationships can be found at:

<http://www.hl7.org/v3ballot/html/infrastructure/datatypes/datatypes.htm>. The documentation of how the datatypes are expressed using XML syntax can be found at:

<http://www.hl7.org/v3ballot/html/infrastructure/itsxml/datatypes-its-xml.htm>.

The CeRx message specification makes use of a subset of the available HL7 datatypes. In many cases, the specification also constrains the datatypes by limiting what properties can be used, restricting the vocabulary, specifying maximum lengths, etc.

This document provides a listing of the datatypes used by CeRx, and also indicates the constraints which have been applied. However, readers are encouraged to reference the underlying HL7 specifications for a more complete explanation of the datatypes and their properties.

It is expected that these materials will ultimately be incorporated into the CeRx Implementation Guide.

## 1.1 Scope

The scope of this document covers the full spectrum of the CeRx standard as well as any associated base models.

## 1.2 Purpose

The purpose of this document is to highlight the data types and constraints on datatypes in use within the specification.

## 1.3 Audience

CeRx-pCSG (Standards Working Group) and other interested stakeholders.

## 1.4 Assumptions

No material assumptions have been made in the preparation of this document.

## 1.5 Related Documents

Author	Document Name	Comments
Michael van Campen	PN502-0001-EN - Standards Development Documentation Framework (SDDF)	Documents all key project deliverables into 5 categories, of which the following 3 are germane to this package: Project Artifacts Package Artifacts Cross-Package Artifacts
CeRx Project Team	PN502-3005-EN - Vocabulary Overview - ;yyyymmdd	This overview document summarizes all vocabulary categories and provides an analysis of options together with recommendations for each code domain.
CeRx Project Team	PN502-3004-EN - Vocabulary Status – yyyymmdd	The vocabulary status workbook provides the primary source of information for all CeRx code domains and the proposed codes or external coding frameworks.

Author	Document Name	Comments
CeRx Project Team	OID Strategy (remains to be published)	Outlines the CeRx approach to assuring identifier uniqueness.
CeRx Core Project Team	Package Documents	Suite of deliverables for the package, contained in a directory structure that corresponds to the 3 major document categories (Project Artifacts, Package Artifacts and Cross-Package Artifacts).



## II GENERAL NOTES

---

### 2.1 Datatype Flavors

For many datatypes, CeRx is proposing the use of particular datatype 'Flavors' which are named constraints on datatypes. These Flavors are generally referenced directly within the CeRx messaging specification. There is presently discussion within HL7 about how to represent in an instance what Flavor of a datatype is in use (as a parallel to `xsi:type` which identifies the international datatype in use). This mechanism will be used within CeRx instances once defined and approved at the international level.

### 2.2 Null Flavor

All HL7 datatypes intrinsically support a "null Flavor" property. If the attribute is not marked as "mandatory" in the message specification, a null Flavor may be specified instead of populating any of the other properties. "Null Flavor" will only appear if the remaining properties are left unspecified.

The null Flavors supported for this specification are:

Code	Print-name
NI	No Information (default)
MSK	Masked
NA	Not Applicable
OTH	Other
NINF	Negative Infinity*
PINF	Positive Infinity*
UNK	Unknown
ASKU	Asked but unknown
NAV	Temporarily unavailable
NASK	Not asked
QS	Sufficient quantity
TRC	Trace

(indentation indicates specialized codes. For example, "ASKU" is a type of "UNK")

\*Negative and positive infinity should only be used for the lower and upper bounds of the Interval (IVL) datatype. They are used to explicitly state "everything lower than" or "everything higher than".

### 2.3 Validation

Many of the restrictions on HL7 v3 datatypes are enforced by HL7 schemas. However, there are some constraints which are not enforced. Many of the CeRx constraints will also not be enforced. Applications claiming conformance with the CeRx specification will be expected to comply with the constraints listed here.

## III COLLECTIONS

---

When a field can repeat, the datatype is shown as either a SET<x>, a LIST<x> or a BAG<x> where x is the datatype of the items that repeat. These “collection wrappers add no additional properties. They merely indicates the characteristics of the collection.

### 3.1 Sets

In sets, all elements in the collection must be unique and order is irrelevant. I.e. the receiver doesn't need to pay attention to the order of the elements.

### 3.2 Lists

In lists order matters and duplicates are allowed. The specific meaning of the order of the elements is conveyed by the definition of the attribute.

## IV CODES AND IDENTIFIERS

---

### 4.1 II - Installer Identifier

This datatype is used for all identifiers, including prescription order identifiers, prescription dispense identifiers, allergy identifiers and application identifiers. It uses a combination of an OID or GUID-based root with an alpha-numeric extension to convey globally unique identifiers.

E.g. <identifier root="1.3.6.1.4.1.12009.24.387" extension="123A45"/>

#### 4.1.1 Root

This component is constrained to be an ISO OID (a registered hierarchical identifier consisting of a sequence of integers separated by periods) or a UUID (a 32-character hex identifier). This component forms the namespace for the issued identifiers. For example, there would be a single ISO OID which represents Alberta-issued Prescription order numbers. A different ISO OID would represent B.C.-issued provincial health identifiers.

This component is mandatory (cannot be null) and has an upper limit of 100 characters.

Refer to the CeRx OID strategy document for information about what OIDs to use for different types of identifiers.

#### 4.1.2 Extension

This component contains the specific identifier. For example, the PHN, the prescription number, *etc.*

The component is required and has a maximum length of 20 characters. For some identifier types, this component will not be used. For other identifier types, the component is mandatory. Again, refer to the CeRx OID strategy document for information about how to use the extension property for different identifier types.

### 4.2 CS - Coded Simple

This datatype is used for a number of internal HL7 attributes which help to define the message structure. These are generally constrained to a fixed value within the specification and are not displayed to users.

The "displayName" property is prohibited for use in the CeRx specification.

E.g. < . . . classCode= "OBS" . . . />

#### 4.2.1 Code

This is the value of the code being conveyed. No code system is specified because the only permitted code-system is HL7-defined. Local codes are prohibited.

The length limit of this mandatory component is 20 characters.

### 4.3 Coded Value

This datatype is used to convey most CeRx coded values, such as type of observation, type of dispense, person gender, *etc.* The behavior of this datatype differs slightly depending on whether the attribute using the CV datatype was marked as CNE (coded with no extensibility) or CWE (coded with extensibility).

The properties "codeSystemName", "codeSystemVersion" and "displayName" merely provide redundant information and are prohibited for use in the CeRx specification.

E.g. <administrativeGender code="F" codeSystem="2.16.840.1.113883.5.1">

```
<originalText>Female</originalText>
</administrativeGender>
```

### **4.3.1 code**

This is the value of the code being conveyed. For example, for gender it might be “F”. For a LOINC code, it might be “10157-6”.

The component is mandatory for “CNE” attributes and required for “CWE” attributes. It has a maximum length of 20 characters. However, implementers are free to restrict the length stored based on the maximum length limit of the code-system in use. For example, when storing LOINC codes, only 7 characters are required.

### **4.3.2 codeSystem**

This is the OID assigned to the code system when it was registered with HL7. The OIDs of the code-systems used for the CeRx specification are listed in the Vocabulary Tracking Framework document.

This component is mandatory if “code” is specified, and is not-permitted otherwise. The maximum length of this component is 100 characters.

### **4.3.3 originalText**

For CNE attributes, originalText is only specified if the “code” property is present. It indicates the text which was seen by the user in selecting the code. For example, if the code were “F”, the originalText might be “Female” (the print name for the code).

For CWE attributes, originalText has the same characteristic if “code” is present. However, originalText may also be included if “code” is not present (i.e., no appropriate code was available). In this case, “originalText” represents what was actually typed by the user in place of selecting an appropriate code.

The component is Required (must be supported but may be null) and has a maximum length of 150 characters.

# V TEXT

---

## 5.1 ST - String

This is a standard string.

For the purposes of this specification, the character-set will be fixed as 128-bit ASCII. The “language” property will not be specified and will be inferred as the default language specified for the message, or failing that for the jurisdiction from which the message originates will apply (e.g. “eng” for English or “fre” for French.)

E.g. <name>Royal Alexandra Hospital</name>

## 5.2 ED.DOCORREF – Encapsulated Data (Document or Reference)

This allows for the encapsulation of a document or reference to a document within the CeRx HL7 instance. In CeRx, this datatype is used to convey monographs, protocols and similar documents.

The properties “charset”, “integrityCheck”, “reference” and “thumbnail” are not supported for this restriction on the ED datatype. There are two alternative sets of properties that can be transmitted. If sending an actual document, then compression, language, mediaType and content are included. If sending a reference, then language, mediaType and reference are included.

E.g. <text mediaType="text/html"><html> . . . </html></text>

E.g. <text mediaType="application/pdf" reference="ftp://monographs.somewhere.gov.ca/12345.pdf"/>

### 5.2.1 compression

This coded component is not mandatory. It may only be populated if content is present. If specified, this indicates the mechanism used to compress the attached document. Only one compression mechanism is supported – GZ (gzip)

### 5.2.2 language

This coded component is not mandatory. If specified, it indicates the language in which the document is written. If not specified, the default language specified for the message, or failing that for the jurisdiction from which the message originates will apply (e.g. “eng” for English or “fre” for French.)

### 5.2.3 mediaType

This mandatory coded component indicates the type of attachment. For the ED.DOC restriction on the ED datatype, there are only 4 permitted values: “text/plain” (for plain-text documents), “text/html” (for HTML documents), “text/xml” (for embedded HL7 CDA documents) and application/pdf (for PDF documents).

Code	Print-name
text/plain	Plain text
text/html	HTML
text/xml	XML (for CDA documents)
application/pdf	PDF

### 5.2.4 reference

This provides a web location from which the document can be retrieved. It is conditional. It may only be populated if content is not present. Supported protocols are FTP, HTTP and HTTPS.

### 5.2.5 <content>

This component represents the content of the document. It must be specified if reference is not specified, and cannot be specified otherwise. If the content is compressed (the compression attribute is populated), or if the content is a PDF document (mediaType="application/pdf", then the content will be MIME-encoded. Otherwise, the content will be sent as directly embedded text or XML. (NOTE: This means that uncompressed HTML must be XHTML compliant.) Content is restricted to a maximum of 1 megabyte after compression and MIME encoding.

## 5.3 ED.REF – Encapsulated Data (Reference)

This allows for a reference to a document within the HL7 instance. In CeRx, this datatype is used to convey references to monographs, protocols and similar documents.

The properties "charset", "compression", "integrityCheck", "reference", "thumbnail" and "content" are not supported for this restriction on the ED datatype.

E.g. <text mediaType="application/pdf" reference="ftp://monographs.somewhere.gov.ca/12345.pdf"/>

### 5.3.1 language

This coded component is not mandatory. If specified, it indicates the language in which the document is written. If not specified, the default language specified for the message, or failing that for the jurisdiction from which the message originates will apply (e.g. "eng" for English or "fre" for French.)

### 5.3.2 mediaType

This mandatory coded component indicates the type of attachment. For the ED.DOC restriction on the ED datatype, there are only 4 permitted values: "text/plain" (for plain-text documents), "text/html" (for HTML documents), "text/xml" (for embedded HL7 CDA documents) and application/pdf (for PDF documents).

Code	Print-name
text/plain	Plain text
text/html	HTML
text/xml	XML (for CDA documents)
application/pdf	PDF

### 5.3.3 reference

This provides a web location from which the document can be retrieved. It is conditional. It may only be populated if content is not present. Supported protocols are FTP, HTTP and HTTPS.

## 5.4 ED.SIGNATURE – Encapsulated Data (Signature)

This allows for the communication of digital signatures within the CeRx HL7 instance. The specific format for the signature is defined by the [W3C XML-Signature Recommendation](#). Further work on how the signature would be applied remains to be undertaken and is expected to be driven by regulatory requirements.

The properties "charset", "compression", "language", "integrityCheck", "reference" and "thumbnail" are not supported for this restriction on the ED datatype.

E.g.

```
<text mediaType="text/xml">  
  <signature xmlns="http://www.w3.org/2000/09/xmldsig#"> . . . </Signature>
```

</text>

#### **5.4.1 mediaType**

This mandatory coded component must be populated with the value “text/xml”.

#### **5.4.2 <content>**

This component contains the XML digital signature as defined in the w3c digital signature specification.

## VI TIMING

---

### 6.1 TS.DATE – Timestamp (Date, partial allowed)

This constraint on the timestamp datatype only permits the “date” portion of the timestamp to be specified. It also allows for partial dates to be transmitted. The grammar for the datatype is: YYYY[MM[DD]] (4-8 characters).

Dates (or portions) specified must be valid dates. (I.e. month must be 01 to 12, February 29 is only permitted for leap-years, etc.)

E.g. <effectiveTime value=“200508”/>

### 6.2 TS.FULLDATE – Timestamp (fully-specified date only)

This constraint on the timestamp datatype only permits the “date” portion of the timestamp to be specified. It does not allow for partial dates to be transmitted. The grammar for the datatype is: YYYYMMDD (8 characters).

Dates specified must be valid dates. (e.g. month must be 01 to 12, February 29 is only permitted for leap-years, etc.)

E.g. <effectiveTime value=“20050803”/>

### 6.3 TS.FULLDATETIME – Timestamp (fully-specified date and time only)

This constraint on the timestamp datatype requires both the “date” and “time” portion of the timestamp to be specified. It does not allow for partial dates or times to be transmitted. The grammar for the datatype is: YYYYMMDDhhmmss (14 characters).

Dates and times specified must be valid dates times. (e.g. month must be 01 to 12, February 29 is only permitted for leap-years, hour must be between 00 and 23, etc.)

E.g. <effectiveTime value=“20050803173000”/>

### 6.4 IVL.LOW<TS. DATE> – Start Date

This datatype is used when both a start and end date may exist, but only the start date should be captured. The interval datatype is used, but only the “low” (start) property is populated. The other properties (“high”, “width”, “center”, “low-closed” and “high-closed”) are not permitted.

The value of the high property is be sent as a fully or partially-specified date. I.e. Either Year, Year + month or complete date.

E.g. “Began July, 2005” would be specified as:

```
<effectiveTime xsi:type="IVL_TS">  
  <low value= "200507"/>  
</effectiveTime>
```

### 6.5 IVL.LOW<TS. FULLDATE> – Start Date (complete)

This datatype is used when both a start and end date may exist, but only the start date should be captured. The interval datatype is used, but only the “low” (start) property is populated. The other properties (“high”, “width”, “center”, “low-closed” and “high-closed”) are not permitted.

The value of the high property is be sent as a fully-specified date. I.e. Partial dates are not allowed.

E.g. “Began July 6, 2005” would be specified as:



```
<effectiveTime xsi:type="IVL_TS">
  <low value="20050706"/>
</effectiveTime>
```

## 6.6 IVL.HIGH<TS.FULLDATE> – End Date

This datatype is used when both a start and end date may exist, but only the end date should be captured. The interval datatype is used, but only the “high” (end) property is populated. The other properties (“low”, “width”, “center”, “low-closed” and “high-closed”) are not permitted.

The value of the high property is sent as a fully-specified date. I.e. Partial dates are not allowed.

E.g. “Ended August 3, 2005” would be specified as:

```
<effectiveTime xsi:type="IVL_TS">
  <high value="20050803"/>
</effectiveTime>
```

## 6.7 IVL<TS. DATE> – Uncertain date range

This datatype is used when a date range needs to be conveyed where neither the start nor end date will always be known to the complete date. The properties (“low-closed” and “high-closed”) are not permitted.

E.g. “from March 01, 2000 to July 2000” would be specified as:

```
<effectiveTime xsi:type="IVL_TS">
  <low value="20000301"/>
  <high value="200007" />
</effectiveTime>
```

E.g. “April, 2004 +/- 2 months” would be specified as:

```
<effectiveTime xsi:type="URG_TS">
  <center value="200404"/>
  <width value="4" unit="m"/>
</effectiveTime>
```

### 6.7.1 low

This indicates the lower boundary-date for the range. It may be specified as a full or partial date. The grammar for the datatype is: YYYY[MM[DD]] (4-8 characters).

### 6.7.2 high

This indicates the upper boundary-date for the range. It may be specified as a full or partial date. The grammar for the datatype is: YYYY[MM[DD]] (4-8 characters).

### 6.7.3 center

This indicates the middle of the date range. It may be specified as a full or partial date. The grammar for the datatype is: YYYY[MM[DD]] (4-8 characters).

### 6.7.4 width

This indicates the total duration of the range. It is sent as a physical quantity (PQ) where the units are restricted to time. The allowed unit codes are:

Code	Print-name
d	Days

wk	Weeks
mo	Months
a	Years

## 6.8 IVL<TS. FULLDATE> – Date range

This datatype is used when a date range needs to be conveyed where the start and end date must be fully specified if communicated at all. The properties (“low-closed” and “high-closed”) are not permitted.

E.g. “from March 01, 2000 to July 31, 2000” would be specified as:

```
<effectiveTime xsi:type="IVL_TS">
  <low value="20000301"/>
  <high value="20000731" />
</effectiveTime>
```

E.g. “April 15, 2004 +/- 2 months” would be specified as:

```
<effectiveTime xsi:type="URG_TS">
  <center value="20040415"/>
  <width value="4" unit="m"/>
</effectiveTime>
```

### 6.8.1 low

This indicates the lower boundary-date for the range. It must be specified as a full date (YYYYMMDD).

### 6.8.2 high

This indicates the upper boundary-date for the range. It must be specified as a full date (YYYYMMDD).

### 6.8.3 center

This indicates the middle of the date range. It must be specified as a full date (YYYYMMDD).

### 6.8.4 width

This indicates the total duration of the date range. It is sent as a physical quantity (PQ) where the units are restricted to time. The allowed unit codes are:

Code	Print-name
d	Days
wk	Weeks
mo	Months
a	Years

## 6.9 IVL.WIDTH<TS. FULLDATE> – Duration

This datatype is used when there is a need to communicate the duration of a date range but not the start or end points. The other properties (“high”, “low”, “center”, “low-closed” and “high-closed”) are not permitted.

E.g. “2 weeks” would be specified as:

```
<effectiveTime xsi:type="IVL_TS">
```

```
<width value= "2" unit="wk"/>
</effectiveTime>
```

### 6.9.1 width

This indicates the total duration. It is sent as a physical quantity (PQ) where the units are restricted to time. The allowed unit codes are:

Code	Print-name
d	Days
wk	Weeks
mo	Months
a	Years

## 6.10 GTS.BOUNDEDPIVL – General Timing Specification (Bounded periodic interval)

This datatype is used to convey two pieces of information: The overall time-period when something occurred (or is to occur), as well as how often it should/did occur within that time-period. This is accomplished by sending two repetitions of the attribute. The first repetition has a type of IVL<TS.FULLDATE> and specifies an operator of "I" (Include). The second repetition has a type of PIVL.

E.g. "3 times/day beginning August 3rd for 3 weeks" would be specified as:

```
<effectiveTime xsi:type= "SXPRTS">
  <comp xsi:type="IVL_TS" operator="I">
    <low value= "20050803"/>
    <width value="3" unit="wk"/>
  </comp>
  <comp xsi:type="PIVL_TS">
    <frequency>
      <numerator value="3"/>
      <denominator value="1" unit="d"/>
    </frequency>
  </comp>
</effectiveTime>
```

# VII MEASUREMENT

---

## 7.1 INT.NONNEG – Integer (Non-negative)

Used to communicate whole numbers greater or equal to 0. Maximum length is restricted to 10 digits.

E.g. <sequenceNumber value="27"/>

## 7.2 INT.POS – Integer (Positive)

Used to communicate whole numbers greater than 0. Maximum length is restricted to 10 digits.

E.g. <sequenceNumber value="27"/>

## 7.3 PQ.BASIC – Physical Quantity (Basic)

Used to communicate measurements using common measurement units encountered in healthcare.

E.g. <value value="22.35" unit="mmol/mL"/>

### 7.3.1 value

This mandatory property indicates the numeric amount of the measurement. The format of the amount is 99999999.99 with no leading or trailing zeros.

### 7.3.2 unit

Indicates how the amount was measured. This coded property must be specified unless the units of measurement are "eaches". For example, "packages", "packs", "capsules", "tablets", etc. are not sent in the units field. These concepts should be explicit in the code identifying the type of item being measured. For example, if the drug name is "100mg Acetaminophen tablets", the quantity would be "1" (or "2" or whatever) with no units specified. The unit would be inferred from the drug name as "tablets".

The codes may be any combination of the atomic units specified in the HL7 "table of units and measure – case sensitive". The list of codes in the Vocabulary Status spreadsheet under the x\_BasicUnitsOfMeasure domain defines the subset of those codes which are expected to be used for CeRx.

## 7.4 PQ.DRUG – Physical Quantity (drugs)

This is identical to the PQ.BASIC (Physical Quantity - Basic) datatype except that the units are restricted to those found in the Vocabulary Status spreadsheet under the x\_DrugUnitsOfMeasure domain.

## 7.5 PQ.HEIGHTWEIGHT – Physical Quantity (height or weight)

This is identical to the PQ.BASIC (Physical Quantity - Basic) datatype except that the units are restricted to those found in the Vocabulary Status spreadsheet under the x\_HeightOrWeightUnitsOfMeasure domain.

## 7.6 PQ.TIME – Physical Quantity (time)

This is identical to the PQ.BASIC (Physical Quantity - Basic) datatype except that the units are restricted to those found in the Vocabulary Status spreadsheet under the x\_TimeUnitsOfMeasure domain.

## 7.7 IVL<PQ.x> - Physical Quantity Range

This is used to express an allowed range of quantities. It may be used with any of the Flavors of PQ datatype (i.e. PQ.BASIC, PQ.DRUG, PQ.TIME, etc.) with the resulting constraint on the allowed units of measure. The properties lowClosed, highClosed, center and width are not supported.

E.g. <value xsi:type="IVL\_PQ">  
    <low value= "500" unit="mg"/>  
    <high value= "2.34" unit="g"/>  
</value>

## 7.8 URG<PQ.x> - Physical Quantity Uncertain Range

This is used when the physical quantity isn't certain or is allowed to vary. It may be used with any of the Flavors of PQ datatype (i.e. PQ.BASIC, PQ.DRUG, PQ.TIME, etc.) with the resulting constraint on the allowed units of measure. The properties lowClosed, highClosed, center and width are not supported.

E.g. <doseQuantity xsi:type="URG\_PQ">  
    <low value= "10" unit= "mg"/>  
    <high value= "4" unit="mg"/>  
</doseQuantity>

## 7.9 RTO<PQ.DRUG, PQ.TIME> Drug Quantity Per Time

This is used to communicate a quantity of a drug over a period of time. The specification for the numerator is as described by PQ.DRUG and the denominator by PQ.TIME. There is a constraint that if the datatype is not null, both numerator and denominator must be non-null.

E.g. <maxDoseQuantity xsi:type= "RTO\_PQ\_PQ">  
    <numerator xsi:type="PQ" value= "200" unit= "mg"/>  
    <denominator xsi:type="PQ" value= "1" unit="d"/>  
</maxDoseQuantity>

# VIII DEMOGRAPHIC

---

## 8.1 PN.BASIC

Used to communicate person names for communication purposes. This datatype is not sophisticated enough for registry-type purposes.

The name parts are limited to given, family, prefix and suffix. The “use” attribute (which applies to the entire name) and “qualifier” attribute (which applies to each name part) are restricted to the following values:

### 8.1.1 Name use

Code	Print-name
L	Legal
P	Pseudonym

### 8.1.2 Name qualifier

Code	Print-name
IN	Initial

The maximum number of name parts is limited to 7. Each name part is limited to a maximum of 30 characters. The validTime attribute is not permitted

E.g. <name use="L"><prefix>Mr.</prefix> <given>John</given> <given qualifier="IN">W.</given> <family>Smith</family></name>

Names may also be expressed in a simpler format without part identification.

E.g. <name use="L">Mr. John W. Smith</name>

Systems are expected to be able to support names entered or transmitted in either format, though discrete name parts should be captured whenever feasible.

## 8.2 TN

Used for drug names.

## 8.3 AD.BASIC

Used to communicate addresses for contact purposes. This datatype is not sophisticated enough for address-registry-type purposes.

The address parts are restricted to city, state (province), zip code (postal code) and country. All other address lines will be specified as free-text with each line demarked by a delimiter address part. A maximum of 4 delimited address lines may be specified in addition to the other address parts.

For country and state address parts, coded values may also be specified, restricted to ISO 3166-1 2-character alpha codes and ISO 3166-2 code suffixes respectively. (For example, Canada would be “CA” and British Columbia would be “BC”. These are the same codes used by Canada Post). . The complete listing of codes may be found here: [http://en.wikipedia.org/wiki/ISO\\_3166-1](http://en.wikipedia.org/wiki/ISO_3166-1).

A maximum of 3 address use codes may be specified from the following list:

Code	Print-name
------	------------

PHYS	Visit address
PST	Postal address
TMP	Temporary address
H	Home
WP	Work place

The length of each address part or content between address parts is restricted to 80 characters, including each address line demarked by a delimiter. The usablePeriod attribute is not permitted.

E.g. <addr use="H PST">Some Company<delimiter/>Apt A5, 123 Some Street N.W.<delimiter/><city>Edmonton</city>, <state code="AB">Alberta</state> <postalCode>A1B 2C3</postalCode><delimiter/><country code="CA">Canada</country></addr>

## 8.4 TEL.PHONEMAIL

This restriction on the TEL datatype only permits telephone and fax numbers as well as e-mail addresses to be communicated. The telecommunication use code may repeat up to three times and is restricted to the following list:

Code	Print-name
H	Home
WP	Work place
EC	Emergency contact
PG	Pager (not for use with e-mail)
MC	Mobile Contact
TMP	Temporary

The telecommunication schemes permitted are:

Code	Print-name
Tel	Telephone
Fax	Work Fax
mailto	E-mail

For telephone and fax numbers, the format is described by <http://www.ietf.org/rfc/rfc2806.txt>. The maximum length is 25 characters for phone and fax numbers and 50 characters for e-mail addresses.

The usablePeriod property is not permitted.

E.g. <telecom use="WP" value="mailto://someone@somewhere.com"/>

## 8.5 TEL.URI

This restriction on the TEL datatype only computer system addresses to be communicated.

The telecommunication schemes permitted are:

Code	Print-name
File	File
ftp	FTP
http	HTTP

<b>Code</b>	<b>Print-name</b>
https	HTTPS
mailto	E-mail
Nfs	NFS

The specific formats of the URIs is defined by their respective RFC specifications.. The maximum length for URI strings is 255 characters.

The use and usablePeriod properties are not permitted.

E.g. <telecom value="mailto://somesystem@somewhere.com"/>



# IX OTHER DATATYPES

---

## 9.1 BL

Used to communicate a value that may be true or false. Like all other datatypes, nullFlavor is also supported.

E.g. `<negationInd value="true"/>`